

REMARKS

Claims 1-27 are pending. Claim 1 has been amended and new claims 26 and 27 have been added.

Claim Rejections under 35 U.S.C. § 103(a)

Claims 1-24 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,247,032 to Bernardo et al. (hereinafter referred to as *Bernardo*) and further in view of U.S. Patent 6,012,087 to Freivald et al. (hereinafter referred to as *Freivald*) and U.S. Patent 6,560,639 to Dan et al. (hereinafter referred to as *Dan*). Applicants' respectfully traverse these rejections as discussed below.

Bernardo discloses a tool for creating Web sites, where the tool facilitates the creation of a customized Web site without requiring a Web site creator to write or edit HTML code. *Col 4, lines 65-67*. The tool enables a Web site creator to select from among a plurality of features and options to include within a Web site. The tool further comprises a library of templates (e.g., text, fields, HTML code and formulas) that correspond to the available features and options. The templates comprise databases which may include fields, forms, views, text, formulas and profiles that enable customization of the features. A profile may comprise fields, some of which may be designated as required fields. For example, one feature of a Web site may be a list of site areas. One area may be "Company Information." A template corresponding to this feature may include certain text and HTML formatting components for a Web page for this area, with fields for company name, address, contact person, etc. Based on input supplied by the Web site creator, this information may be included in the finished Web site without requiring the Web site creator to write or edit HTML code. *Col. 6, lines 1-16*.

Bernardo does not discuss the use of time stamps in connection with the templates, or the use of dependency records.

Freivald discloses detection of changes in web pages using history tables of signatures. According to *Freivald*, a user may only be interested in differences in the file, or learning when the file is updated, such as when a new product or service is announced. A software tool automatically retrieves files and compares the retrieved files to an archived signature of the file to determine if a change in the file has occurred. When a change is detected, the user is notified by an electronic mail message (e-mail). A copy of the new file may be attached to the e-mail notification, allowing the user to review the changes. *Col. 6, lines 17-25*. Rather than archive the source files from remote document server 12, *Freivald* archives a checksum CRC or signature of the source files. These signatures and the e-mail address of the user are stored in database 16 of change-detection server 20. Comparison is made of the stored or archived signature of the document and a fresh signature of the currently-available document. The signature is a condensed checksum or fingerprint of the document. Any change to the document changes the signature. *Col. 6, lines 26-34*.

Freivald at least does not disclose the use of dependency records, as set forth for example, in claims 1, to determine out of date files.

Dan discloses a method and/or system that allows a user to create and review changes to a web page directly on the World Wide Web and/or in real-time. *Col. 2, lines 46-49*. *Dan* merely provides an authorized user with a webpage attributes form for making proposed edits to the webpage to which the attributes correspond. *Dan* does not describe or suggest any comparison of these attributes with other parameters prior to entering the user's edits or changes to the database at the Internet Service Provider (ISP). In *Dan*, a web management system (Fig.

2) is provided that includes a file system caching all web pages in a web site, whereby the web pages so cached may be at least partially static. The system also includes a web server communicating with the file system. The web site management system may optionally include a database having a directory structure associating each page or web page of a site or web site with attributes thereof. The system may optionally include a server-side front end daemon communicatable with the web server and the database. The server-side front end or other part of the system may identify the attributes of any user-changed page or web page and/or store the attributes of any user-changed page or web page in the database. *Col. 2, lines 54-67.*

In *Dan*, the web management system may further include a server-side back end daemon communicatable with the database and the file system. The back end daemon may simply parse the attributes to generate at least partially static web pages and may store the generated web pages in the file system. *Col. 3, lines 1-6.* An illustrative method of operation of *Dan*'s web management system (Fig. 2) is shown in Fig. 3. In Step S10, it is determined whether the user has requested a web page attributes form from the front end daemon 35 via the web server 20. In Step S20, the front end daemon 35 reads the database 50 associating web page attributes and web pages in a given web site and sends the requested form having the attribute associations to the user via the web server 20 and the user's web browser 10. In Step S30, it is determined whether the user has edited the requested form and submitted this form to the front end daemon 35 via the web server 20. In Step S40, it is determined whether the user is authorized to make the proposed edits. In Step S50, the front end daemon 35 enters the user's changes to the web site to the database 50. In Step S60, the front end daemon 35 calls the back end daemon 40 to parse the edited web page or all of the web pages in the web site and cache the edited web page in the file

system 45. In Step S70, the user, via the web server 10, reads the edited, cached web page from the file system 45.

As the discussion above demonstrates, *Dan* at least does not disclose the use of dependency records, as set forth for example, in claims 1, to determine out of date files.

Claims 1-14

Claim 1 of the application recites a system for content management that automatically determines when a content page contains out of date content items as a result of changes made to the content items in a data source. The system includes a template engine for executing templates to generate a content page, whereby the template engine generates a content page comprising content items selectively retrieved from a data source and arranged on the content page as defined by the template. Each content item in the data source is associated with time stamp information to indicate the last time the content item was modified. The system further includes a dependency record for storing information regarding a relationship between content items that comprise the content page and the content items stored in the data source and time parameter information associated with the content items that comprise the content page. Dependency checking software compares information contained in the dependency record with time stamp information contained in the data source for each content item included in the content page, and determines through the comparison, those content pages that contain content items that have been modified in the data source. The template engine is then instructed to re-generate a content page that contains modified content items.

As stated by the Examiner, neither *Bernardo* nor *Freivald* describes or suggests “a dependency record for storing information regarding a relationship between content items that comprise the content page and the content items stored in the data source.” The Examiner

contends that this deficiency is found in *Dan*. However, *Dan* also fails to describe or suggest “a dependency record for storing information regarding a relationship between content items that comprise the content page and the content items stored in the data source.” In *Dan*, a webpage attributes form is used by a user to make changes to a webpage. The front end daemon 35 merely reads the database 50 associating web page attributes and web pages in a given web site and sends to the user via the web server 20 and the user’s web browser the requested webpage attributes form with these attribute associations. If the user is authorized to make edits to the requested webpage attributes form received by the user’s browser, such edits are entered into the database 50 by the front end daemon 35. *Dan* does not describe or suggest a “dependency record” much less a “dependency record” that “stor[es] information regarding a relationship between content items that comprise the content page and the content items stored in [a] data source.” *Dan* provides a user requested form that utilizes webpage attributes of a webpage to make user-changes to a website. *Dan* does not describe or suggest any comparison or relationship between these attributes and other content items stored in the database prior to entering the user’s edits or changes within the database. Moreover, in *Dan*, once the database is updated, the user displays the edited and cached webpage via the user’s web browser. Thus, *Dan* does not “re-generate a content page that contains modified content items” based on “comparing information contained in the dependency record.” Therefore, for at least these reasons, Applicants respectfully submit that claim 1 is patentable over *Bernardo*, *Freivald*, and *Dan*.

Claims 2-14 and 26-27 depend directly and indirectly from, and include all the subject matter of, claim 1, which is allowable for the reasons discussed above. Therefore, it is submitted that claims 2-14 and 26-27 are also allowable, since they depend from allowable claim 1.

Claims 15-23

Claim 15 of the application recites a method for determining when a content page contains out of date content items as a result of changes made to a template or content items stored in a data source. The method includes generating a template to define the content items to be included in the content page and the arrangement of the content items on the content page. The method further includes executing the template with a template engine to generate the content page as defined by the template and generating one or more dependency records to capture a relationship between the content items that are included in the content page, the template used to generate the content page, and the content items stored in the data source. Data contained in the dependency records is then compared with data contained in the data source to determine if the content page is out of date.

As indicated by the Examiner, neither *Bernardo* nor *Freivald* describes or suggests “generating one or more dependency records to capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” The Examiner contends that this deficiency is found in *Dan*. However, Applicants respectfully submit that *Dan* also fails to describe or suggest “generating one or more dependency records to capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” In *Dan*, a webpage attributes form is used by a user to make changes to a webpage. The front end daemon 35 merely reads the database 50 associating web page attributes and web pages in a given web site and sends to the user via the web server 20 and the user's web browser the requested webpage attributes form with these attribute associations. If the user is authorized to make edits to the requested webpage attributes form received by the

user's browser, such edits are entered into the database 50 by the front end daemon 35. *Dan* does not describe or suggest "generating one or more dependency records" much less "dependency records" that "capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source." As previously described, *Dan* provides a user requested form that utilizes webpage attributes of a webpage to make user-changes to a website. *Dan* does not describe or suggest capturing any comparison or relationship between these attributes and other content items stored in the database prior to entering the user's edits or changes within the database. Moreover, in *Dan*, once the database is updated, the user displays the edited and cached webpage via the user's web browser. Thus, *Dan* does not "determine if the content page is out of date" based on "comparing data contained in the dependency records." Therefore, for at least these reasons, Applicants respectfully submit that claim 15 is patentable over *Bernardo*, *Freivald*, and *Dan*.

Claims 16-23 depend directly and indirectly from, and include all the subject matter of, claim 16, which is allowable for the reasons discussed above. Therefore, it is submitted that claims 16-23 are also allowable, since they depend from allowable claim 15.

Claim 24

Claim 24 of the application recites a method for generating one or more dependency records that can be used to determine when content items that comprise a content page have been modified. The method includes generating a template to define the content items to be included in the content page and the arrangement of the content items on the content page, where the content page includes content items stored in a data source. The method further includes executing the template with a template engine to generate the content page as defined by the template. One or more dependency records are generated to capture a relationship between the

content items that are included in the content page, the template used to generate the content page, and the content items stored in the data source.

As indicated by the Examiner, neither *Bernardo* nor *Freivald* describes or suggests “generating one or more dependency records to capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” Applicants respectfully submit that *Dan* also fails to describe or suggest “generating one or more dependency records to capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” In *Dan*, a webpage attributes form is used by a user to make changes to a webpage. The front end daemon 35 merely reads the database 50 associating web page attributes and web pages in a given web site and sends to the user via the web server 20 and the user’s web browser the requested webpage attributes form with these attribute associations. If the user is authorized to make edits to the requested webpage attributes form received by the user’s browser, such edits are entered into the database 50 by the front end daemon 35. *Dan* does not describe or suggest “generating one or more dependency records” much less “dependency records” that “capture a relationship between the content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” *Dan* merely provides a user requested form that utilizes webpage attributes of a webpage to make user-changes to a website. *Dan* does not describe or suggest capturing any comparison or relationship between these attributes and other content items stored in the database prior to entering the user’s edits or changes within the database. Therefore, for at least these reasons, Applicants respectfully submit that claim 24 is patentable over *Bernardo*, *Freivald*, and *Dan*.

Claim Rejections under 35 U.S.C. § 103(a)

Claim 25 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over *Freivald* in view of *Dan*. Applicants' respectfully traverse this rejections as discussed below.

Claim 25 of the application recites a method for determining when a content page contains content items that are out of date, whereby the content page is generated by instructions contained in a template that identifies content items stored in a data source for inclusion in the content page. The method comprises storing one or more dependency records to capture a relationship between content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source. The method further includes comparing the data contained in the dependency records with data contained in the data source to determine if the content page is out of date. The content page is then regenerated, whereby the comparison step determines that the content page contains modified content items.

Neither *Freivald* nor *Dan* describes or suggests “storing one or more dependency records to capture a relationship between content items that comprise the content page, the template used to generate the content page, and the content items stored in the data source.” As discussed above, in *Dan*, a webpage attributes form is used by a user to make changes to a webpage. The front end daemon 35 merely reads the database 50 associating web page attributes and web pages in a given web site and sends to the user via the web server 20 and the user's web browser the requested webpage attributes form with these attribute associations. If the user is authorized to make edits to the requested webpage attributes form received by the user's browser, such edits are entered into the database 50 by the front end daemon 35. *Dan* does not describe or suggest “storing one or more dependency records to capture a relationship between content items that

comprise the content page, the template used to generate the content page, and the content items stored in the data source.” *Dan* provides a user requested form that utilizes webpage attributes of a webpage to make proposed user-changes to a website. *Dan* does not describe or suggest capturing any comparison or relationship between these attributes and other content items stored in the database prior to entering the user’s edits or changes within the database. Moreover, in *Dan*, once the database is updated, the user displays the edited and cached webpage via the user’s web browser. Thus, *Dan* does not describe or suggest “re-generat[ing] the content page” based on “the comparison step.” Therefore, for at least these reasons, Applicants respectfully submit that claim 25 is patentable over *Freivald* and *Dan*.

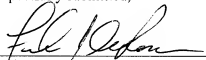
Closing

For at least the reasons stated above, Applicants respectfully submit that all pending claims 1-27 are allowable over the prior art, and allowance of all claims is respectfully solicited. To expedite prosecution of this application to allowance, the Examiner is invited to call the Applicant’s undersigned representative to discuss any issues relating to this application.

Date: 12/29/06

Customer No. 29858
Brown Raysman Millstein Felder & Steiner LLP
900 Third Avenue
New York, NY 10022
Tel. (212) 895-2000 Ext. 2006
Fax (212) 895-2900

Respectfully submitted,


Frank J. DeRosa,
Reg. No. 26,543
Attorney for Applicants